ACTIONS, PROCESSES, AND CAUSALITY

Technical Note 404

February 1987

By: Michael P. Georgeff
Program Director

Representation and Reasoning Program
Artificial Intelligence Center
Computer and Information Sciences Division
and
Center for the Study of Language and Information
Stanford University

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This paper was presented at the *Workshop on Reasoning about Actions and Plans*, held at Timberline Lodge, Timberline, Oregon, in July 1986. It appears in the volume "Reasoning about Actions and Plans: Proceedings of the 1986 Workshop," published by Morgan Kaufmann Publishers. Los Altos, California, and is reproduced here with their kind permission.

This research has been made possible by a gift from the System Development Foundation, by the Office of Naval Research under Contract N00014-85-C-0251, and by the National Aeronautics and Space Administration, Ames Research Center, under Contract NAS2-12521.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **FEB 1987**	2. REPORT TYPE	3. DATES COVERED **00-02-1987 to 00-02-1987**
4. TITLE AND SUBTITLE **Actions, Processes, and Causality**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**24**	

Abstract

The purpose of this paper is to construct a model of actions and events that facilitates reasoning about dynamic domains involving multiple agents. Unlike traditional approaches, the proposed model allows for the simultaneous performance of actions, rather than use an interleaving approximation. A generalized situation calculus is constructed for describing and reasoning about actions in multiagent settings. Notions of *independence* and *correctness* are introduced, and it is shown how they can be used to determine the persistence of facts over time and whether or not actions can be performed concurrently. Unlike most previous formalisms in both single- and multiagent domains, the proposed law of persistence is *monotonic* and thus has a well-defined model-theoretic semantics. It is shown how the concept of *causality* can be employed to simplify the description of actions and to model arbitrarily complex machines and physical devices. Furthermore, it is shown how sets of causally interrelated actions can be grouped together in *processes* and how this structuring of problem domains can substantially reduce combinatorial complexity. Finally, it is indicated how the law of persistence, together with the notion of causality, makes it possible to retain a simple model of action while avoiding most of the difficulties associated with the frame problem.

1 Introduction

In developing automatic systems for planning and reasoning about actions, it is essential to have epistemologically adequate models of events, actions, and plans. Most early work in action planning assumed the presence of a single agent acting in a static world. In the formulation of these problems, the world was considered to be in one of a potentially infinite number of states and actions were viewed as mappings between these states [4,16,19,23]. However, the formalisms developed did not allow for simultaneous action, and as such are inadequate for dealing effectively with most real-world problems that involve other agents and dynamically changing environments.

Some attempts have recently been made to provide a better underlying theory of action. McDermott [20] considers an action (or event) to be a set of sequences of states and describes a temporal logic for reasoning about such actions. Allen [1] adopts a similar view and specifies an action by giving the relationships among the intervals over which the action's conditions and effects are assumed to hold. Related formalisms have been developed by Dean [3], Pelavin [24] and Shoham [28].

A quite different and potentially powerful approach has recently been proposed by Lansky [14]. Instead of modeling actions and events in terms of world states, she regards events as primitive and defines states derivatively.

In this paper, we shall examine some of the problems that arise in the representation of events, actions, and plans in multiagent domains, and describe a model of events and actions that overcomes most of these problems.

2 Actions and Events

We consider that, at any given instant, the world is in a particular *world state*. Each world state consists of a number of *objects* from a given domain, together with various *relations* and *functions* over those objects. A sequence of world states is called a *world history*.

A given world state has no duration; the only way the passage of time can be observed is through some change of state. The world changes its state by the occurrence of *events* or *actions*.[1] An *event type* is a set of state sequences, representing all possible occurrences of the event *in all possible situations* [1,20]. Except where the distinction is important, we shall call event types simply events.

We shall restrict our attention herein to *atomic events*. An atomic event is one in which each state sequence comprising the event contains exactly two elements; it can thus be modeled as a transition relation on world states. The transition relation of a given event must comprise all possible state transitions, *including those in which other events occur simultaneously with the given event.* Consequently, the transition relation of an atomic event places restrictions on those world relations that are directly affected by the event, but leaves most others to vary freely (depending upon what else is happening in the world).

[1] From a technical standpoint, we shall use these terms synonymously.

This is in contrast to the classical approach, which views an event as changing some world relations but leaving most of them unaltered.

For example, consider a domain consisting of blocks A and B at possible locations 0 and 1. Assume a world relation that represents the location of each of the blocks, denoted loc. Consider two events, $move(A,1)$, which has the effect of moving block A to location 1, and $move(B,1)$, which has a similar effect on block B. According to the classical approach [23], these events would be modeled as follows:

$$\begin{array}{rcl} move(A,1) & = & \{\langle loc(A,0), loc(B,1)\rangle \rightarrow \langle loc(A,1), loc(B,1)\rangle \\ & & \langle loc(A,0), loc(B,0)\rangle \rightarrow \langle loc(A,1), loc(B,0)\rangle\} \end{array}$$

and similarly for $move(B,1)$.

Every instance (transition) of $move(A,1)$ leaves the location of B unchanged, and similarly every instance of $move(B,1)$ leaves the location of A unchanged. Consequently, it is impossible to compose these two events to form one that represents the simultaneous performance of both $move(A,1)$ and $move(B,1)$, except by using some interleaving approximation [22].

In contrast, our model of these events is

$$\begin{array}{rcl} move(A,1) & = & \{\langle loc(A,0), loc(B,1)\rangle \rightarrow \langle loc(A,1), loc(B,1)\rangle \\ & & \langle loc(A,0), loc(B,1)\rangle \rightarrow \langle loc(A,1), loc(B,0)\rangle \\ & & \langle loc(A,0), loc(B,0)\rangle \rightarrow \langle loc(A,1), loc(B,1)\rangle \\ & & \langle loc(A,0), loc(B,0)\rangle \rightarrow \langle loc(A,1), loc(B,0)\rangle\} \end{array}$$

and similarly for $move(B,1)$.

This model represents all possible occurrences of these events, including their simultaneous execution with other events. For example, if $move(A,1)$ and $move(B,1)$ are performed simultaneously, the resulting event will be the intersection of their possible behaviors:

$$\begin{array}{rcl} move(A,1) \| move(B,1) & = & move(A,1) \cap move(B,1) \\ & = & \{\langle loc(A,0), loc(B,0)\rangle \rightarrow \langle loc(A,1), loc(B,1)\rangle\} \end{array}$$

Thus, to say that an event has taken place is simply to place constraints on some world relations, while leaving most of them to vary freely.

Of course, to specify events by listing all the possible transitions explicitly would, in any interesting case, be infeasible. We therefore need some formalism for describing events and world histories. The one we use here is a generalization of the situation calculus [19], although most of our remarks would apply equally to other logic-based formalisms.

We first introduce the notion of a *fluent* [19], which is a function defined on world states. If we are using predicate calculus, the values of these fluents will range over the relations, functions, and objects of the domain. For example, the location of a given block A is a fluent whose value in a given state is the location of block A in that state. If, in a state s, the location of A is 1, we shall write this as $holds(loc(A,1), s)$. Expressions denoting fluents that range over objects are often called *designators*, with a distinction drawn between those whose denotations are constant over all states (so-called *rigid* designators) and those whose denotations may vary (*nonrigid* designators).

As in the single-agent case, the well-formed formulas of this situation calculus may contain logical connectives and quantifiers; they can thus express general assertions about world histories. However, we do not use a "result" function to specify the state resulting from an event (or action). The reason is that, in our formalism, events are not functions on states but rather relations on states, and the occurrence of an event in a given state need not uniquely determine the resulting state. Therefore, for a given world history w containing state s, we let $succ(s, w)$ be the successor of s, and use a predicate $occurs(e, s)$ to mean that event e occurs in state s. This formulation, in addition to allowing a wider class of events than in the standard situation calculus, also enables us to state arbitrary temporal constraints on world histories [24].

In reasoning about actions and events, one of the most important things we need to know is how they affect the world – that is, we must be able to specify the effects of actions and events when performed in given situations. We can do this as follows.

Let ϕ and ψ be relational fluents. Then we can describe the effects of an event e with axioms of the following form:[2]

$$\forall w, s \;.\; holds(\phi, s) \wedge occurs(e, s) \supset holds(\psi, succ(s, w)) \tag{1}$$

This statement is intended to mean that, if ϕ is true when event e occurs, ψ will be true in the resulting state. It has essentially the same meaning as $\phi \supset [e]\psi$ in dynamic logic [11].

With axioms such as these, we can determine the strongest [provable] postconditions and weakest [provable] preconditions of arbitrary events and actions. These can then be used to form plans of action to accomplish given goals under prescribed initial conditions [17,26].

It is important to note that axioms of the above form cannot characterize the transition relation of any given event completely, no matter how many are provided. For example, with such axioms alone, it is not possible to prove for any two actions that they can be performed concurrently (nor that a plan containing concurrent actions is executable). We shall have more to say about this later.

Of course, we are often able to make stronger statements about actions and events than given above. For example, the event $move(A, 1)$ satisfies

$$\forall w, s \;.\; occurs(move(A, 1), s) \equiv holds(loc(A, 0), s) \wedge holds(loc(A, 1), succ(s, w))$$

This specification characterizes the event $move(A, 1)$ completely – there is nothing more that can be said about the event or, more accurately, about its associated transition relation.

[2] We have simplified the notation in two ways. First, without stating so explicitly, we assume throughout that s and $succ(s, w)$ are elements of w. Second, we shall often use event *types* to stand for an event *instance* of the given type. Thus, axiom 1 should be viewed as shorthand for the following axiom, where ι is an event instance:

$\forall w, s, \iota \;.\; element(s, w) \wedge element(succ(s, w), w) \wedge type(\iota, e) \wedge holds(\phi, s) \wedge occurs(\iota, s) \supset holds(\psi, succ(s, w))$.

(The importance of this distinction will be made clear when we consider causal relationships among events.)

Thus, *at this point in the story*, the frame problem [12,19] does not arise. Because events, per se, need not place any restrictions on the majority of world relations, we do not require a large number of frame axioms stating what relations are left unchanged by the performance of an event (indeed, such statements would usually be false). In contrast to the classical approach, we therefore do not have to introduce any *frame rule* [12] or STRIPS-like assumption [4] regarding the *specification* of events.

3 Independence

We have been regarding atomic actions or events as imposing certain constraints on the way the world changes while leaving other aspects of the situation free to vary as the environment chooses. That is, each action's transition relation describes all the potential changes of world state that could take place during the performance of the action. Which transition actually occurs in a given situation depends, in part, on the actions and events that take place in the environment. However, unless we can reason about what happens when some subset of all possible actions and events occurs – for example, when the only relevant actions being performed are those of the agent of interest – we could predict very little about the future and any useful planning would be impossible.

To handle this problem, we first introduce the concept of *independence*. We define a predicate $indep(p, e, s)$, which we take to mean that the fluent p is independent of (i.e., not directly affected by) event e in situation s. Unlike classical models of actions and events, this does not mean that, if we are in a state s in which p holds, p will also hold in the resulting state. Rather, if p is independent of e in some state s, the transition relation associated with e will include transitions to states in which p does not hold, as well as ones in which p holds, while not constraining the values of any other fluents in the resulting state.[3] In the general case, we have to specify independence for all three types of fluents: the relation-valued, function-valued, and object-valued.

For example, we might have

$$\forall s, x, y \, . \, holds(x \neq A, s) \supset indep(loc(x, y), move(A, 1), s) \; .$$

This axiom states that, for all x and y, $loc(x, y)$ will be independent of event $move(A, 1)$, provided that x is not block A.

In our ontology, a world state can change only through the occurrence of events. Furthermore, in keeping with our intuitive notion of independence, events that are independent of some property cannot influence that property. We therefore have

$$\forall w, s, \phi \, . \, holds(\phi, s) \wedge \neg holds(\phi, succ(s, w)) \supset \exists e \, . \, (occurs(e, s) \wedge \neg indep(\phi, e, s)) \qquad (2)$$

[3] Independence can be defined as follows. Let $tr(e, s, s')$ denote that $\langle s, s' \rangle$ is an element of the transition relation associated with event e. Then we have that p is independent of e in state s if and only if, for all ϕ such that ϕ is consistent with both p and $\neg p$, $(\exists s' \, . \, tr(e, s, s') \wedge holds(p \wedge \phi, s')) \equiv (\exists s' \, . \, tr(e, s, s') \wedge holds(\neg p \wedge \phi, s'))$. This, of course, is not computable. Note also that an event transition relation may include states that cannot occur (because of some domain constraint) in any world history.

From this we can directly deduce the following *law of persistence*:

$$\forall w, s, \phi \;.\; holds(\phi, s) \wedge \forall e \;.\; (occurs(e, s) \supset indep(\phi, e, s)) \supset holds(\phi, succ(s, w)) \quad (3)$$

This rule states that, if we are in a state s where some condition ϕ holds, and if all events that occur in state s are independent of ϕ, then ϕ will also hold in the next (resulting) state. For example, we could use this rule to infer that, if $move(A, 1)$ were the only event to occur in some state s, the location of B would be the same in the resulting state as it was in s.

Unlike many other approaches to persistence [3,10,12,20,25,28], the foregoing law is *monotonic*; that is, the law does not involve any nonmonotonic operators or depend on any consistency arguments. Nor is it some fortuitous property of the world in which we live. Rather, it is a *direct consequence* of our notions of event and independence. What makes planning useful for survival is the fact that we can structure the world in a way that keeps most properties and events independent of one another, thus allowing us to reason about the future without complete knowledge of all the events that could possibly be occurring.

At first glance, however, it appears as though we would encounter considerable difficulty in specifying independence, simply on the grounds that it should be ascertainable for each possible fluent/event pair. Indeed, this is hardly surprising, as the foregoing law of persistence is little different in this respect from the original formalisms that gave rise to the frame problem [19].

There are two ways we could deal with this difficulty. One is to remain monotonic and rely on general axioms regarding independence to reduce combinatorial complexity. The other is to apply some nonmonotonic rule or minimization criterion that would allow independence to be specified more succinctly. I discuss the nonmonotonic approach elsewhere [5]; herein, I want to examine briefly the monotonic specification of independence, so as to show that such an approach is – in some cases – a practical alternative.

One way in which the combinatorics can be substantially reduced is by explicitly specifying *all* the events that could possibly affect each fluent.[4] For example, for a given fluent p, we might have axioms such as the following:

$$\forall s, e \;.\; \neg indep(p, e, s) \supset ((e = e_1) \vee (e = e_2) \vee \ldots)$$

or, in its contrapositive form

$$\forall s, e \;.\; \neg((e = e_1) \vee (e = e_2) \vee \ldots) \supset indep(p, e, s)$$

As one would expect that, out of all possible events, there will be relatively few that affect a given fluent, such specifications can reduce considerably the combinatorics of providing separate independence axioms for each fluent/event pair. A minor complication is that, because we allow composite events (such as $e_1 \| e_2$, and that combined with, say, e_3, and so forth), the axioms for independence cannot be quite so simple as given above. However, this presents no serious difficulty.

[4] This is essentially what Lansky does when she defines state predicates in terms of events [14].

A more substantial problem, however, is that this approach requires one to know the effects of *all* actions and events that could possibly occur. That is, such axioms do not allow for the possibility that *unspecified* events could affect the fluents of interest. This approach would therefore seem too strong for many real-world applications, though may be useful in less general contexts.

There are other ways to specify independence, however, that manage to avoid the combinatorial problem, yet do so without banishing unspecified events from the scene and without introducing nonmonotonicity. In particular, it may be possible to provide axioms describing the extent to which various actions and events exert their influence. For example, it may be that events outside a particular region R cannot affect properties inside that region:

$$\forall s, e, \phi \, . \, internal_f(\phi, R, s) \wedge external_e(e, R, s) \supset indep(\phi, e, s)$$

In this way, a single axiom can specify independence for an entire class of fluent/event pairs. In large real-world domains this will invariably lead to a substantial reduction in the combinatorics of the problem. In small blocks worlds, on the other hand, it will not – but writing down all the independence axioms in such a case is not much of a problem either.

4 Interference

If we are interested in constructing plans of action, one of the more important considerations is whether or not the actions constituting such plans are indeed performable. In single-agent planning, this question is quite easily handled by means of explicitly specifying preconditions that guarantee action performability. As we shall see, however, it is much more complex in multiagent domains.

The source of the problem in multiagent planning is that it is not possible to state simple preconditions for each individual action, the satisfaction of which would ensure its performability. In multiagent domains, whether or not an action can be performed will depend not only on the fulfillment of such preconditions, but also on which events or actions may (or are required to) occur simultaneously with the given action; it is, after all, of little use to form a plan that calls for the simultaneous or concurrent performance of actions that are inherently precluded from coexisting.

This problem is far more crucial than it may first appear. In particular, we are not concerned merely with issues of deadlock avoidance. In planning and other forms of practical reasoning, the failure of an action does not necessarily mean that the agent or device performing the action will thereafter be unable to proceed. Rather, such failure is usually taken to mean that the *desired* or *intended* effects of the action have not been achieved. Thus, though true deadlock may occur quite rarely, actions often fail to produce their intended effects because of interference with other, often unanticipated events.

Moreover, much of human planning revolves around the *coordination* of plans of action. Some of this is concerned with synchronizing the activities of agents so that tasks involving more that one agent can be carried out successfully. Such synchronization can be accomplished by specifying explicitly what temporal relations should hold among the activities of the various agents [14,30]. The more difficult problem is to identify interactions among potentially conflicting actions. Indeed, the recognition of possible plan conflicts is considered by some philosophers to be at the heart of rational behavior [2].

One way to specify such constraints on actions and events is to provide explicit axioms stating which events should occur simultaneously and which should not. For example, we could have the axiom

$$\forall s \,.\, \neg(occurs(e_1, s) \wedge occurs(e_2, s))$$

to mean that event e_1 could not occur simultaneously with event e_2. This is exactly the approach employed by Lansky [14] and Pelavin [24]. However, while it seems that the synchronization of actions for cooperative tasks is most naturally expressed directly (that is, by explicitly specifying the required temporal relations between specific actions), it seems unreasonable to require that all possible action *conflicts* also be so specified. For most real-world domains, it is more natural to specify just the effects of actions and to *deduce*, as the need arises, whether or not any two actions will interfere with each another. Furthermore, for domains of any complexity, there are potentially a very large number of actions and events that could interfere with one another. In such cases, the explicit specification of interference would entail severe combinatorial difficulties, although *appropriate* structuring of the problem domain [14] could substantially reduce the combinatorics.

It is desirable, therefore, to be able to determine freedom from conflict for any specified events, given simply a description of the effects of these events upon the world. To do this, we need to prove that the intersection of the transition relations corresponding to the events of interest is nonempty.

At first glance, it appears as if axioms about the effects of events are *all* we really need for determining the possibility or not of event simultaneity. For example, let us assume we have the following axioms describing events e_1 and e_2:

$$\forall w, s \,.\, holds(p, s) \wedge occurs(e_1, s) \supset holds(q_1, succ(s, w))$$

$$\forall w, s \,.\, holds(p, s) \wedge occurs(e_2, s) \supset holds(q_2, succ(s, w))$$

From this we can infer that

$$\forall w, s \,.\, holds(p, s) \wedge occurs(e_1 \| e_2, s) \supset holds(q_1 \wedge q_2, succ(s, w))$$

Nevertheless, it would be unwise to take these axioms as the basis of a plan to achieve $q_1 \wedge q_2$. The reason is that, given these axioms alone (or any others of the same form), it is simply not possible to *prove* that events e_1 and e_2 can occur simultaneously. Nor is it possible to use consistency arguments to justify the assertion that these events can so occur; indeed, whether or not these events can take place simultaneously depends on how they affect other world properties. For example, simultaneity would be impossible if e_1, say, always resulted in r being true while e_2 always resulted in r being false. (Of course, given *sufficient* axioms about the effects of e_1 and e_2, we could, in this case, prove that they *could not* occur together.)

Even if we are given necessary and sufficient conditions for the occurrence of events [1], we are still not out of the woods. For example, consider that events e_1 and e_2 satisfy the following axioms:

$$\forall w, s \;.\; occurs(e_1, s) \equiv holds(p, s) \wedge holds(q_1, succ(s, w))$$

$$\forall w, s \;.\; occurs(e_2, s) \equiv holds(p, s) \wedge holds(q_2, succ(s, w))$$

That is, a necessary and sufficient condition for e_1 having occurred is that p holds at its inception and q_1 holds at its completion; for e_2, q_2 must hold at its completion. But, even in this case, the best we can do is to try to prove that it is *consistent* for these events to occur simultaneously. This is clearly unsatisfactory from a computational standpoint. Furthermore, such reasoning is essentially nonmonotonic; the addition of further axioms may render previously consistent formulas inconsistent and any previous conclusions about possible event simultaneity may have to be withdrawn.

Another alternative is to determine interference by checking mutual independence for every fluent in the domain [7]. The major problem with such an approach is that this determination has to be made for every possible fluent, *including unspecified ones.* For example, despite the fact that two events may be mutually independent with respect to every specified fluent in the domain, there may exist some unspecified fluent for which they are not mutually independent. We are thus required to assume that the explicitly denoted fluents are the *only* ones relevant to the determination of interference.

Just as we wanted to avoid introducing any nonmonotonic operator or consistency criterion into our law of persistence, here also we want to avoid any form of nonmonotonicity. The solution we propose is based on being able to specify conditions under which we can guarantee performability of a given action or event. Such a condition will be called a *correctness condition* and, for a given event e, condition p, and state s, will be denoted $cc(p, e, s)$. The intended meaning of this statement is that any event that does not interfere with (affect) condition p will not interfere with (prevent) the occurrence of event e.[5] In addition, of course, we would need appropriate axioms defining the preconditions for the performance of e, but this is easily handled in the standard manner.

We now introduce the notion of freedom from interference. We shall say that events e_1 and e_2 are *interference-free* in a state s if the following condition holds:

$$\exists \phi, \psi \;.\; cc(\phi, e_1, s) \wedge cc(\psi, e_2, s) \wedge indep(\phi, e_2, s) \wedge indep(\psi, e1, s)$$

This condition will hold if, in state s, events e_1 and e_2 have no direct effect on the same properties of the domain. For example, consider the events $move(A, 1)$ and $move(B, 1)$ described earlier. We have the following:[6]

$\forall s, x, y, X \;.\; holds(x \neq X, s) \supset indep(loc(x, y), move(X, 1), s)$

[5] Correctness conditions can be defined as follows. Let $tr(e, s, s')$ denote that $\langle s, s' \rangle$ is an element of the transition relation associated with event e. Then we have that p is a correctness condition for an event e if and only if, for all ϕ such that both ϕ and $\neg\phi$ are consistent with p, $(\exists s' \;.\; tr(e, s, s')) \wedge holds(p \wedge \phi, s')) \equiv (\exists s' \;.\; tr(e, s, s') \wedge holds(p \wedge \neg\phi, s'))$. As with the definition of independence, this is not computable.

[6] We assume x and y are rigid designators; see reference [17] for a discussion of this issue.

$\forall s, X \,.\, cc(loc(X,1), move(X,1), s)$

If A and B are assumed to denote different objects, it is easy to see that $move(A,1)$ and $move(B,1)$ are interference-free. Note that we have assumed that both A and B can occupy the same location at the same time. If this were not the case, the correctness conditions for $move(A,1)$ and $move(B,1)$ would have to be altered to include this additional constraint. The events would then not be interference-free.

It immediately follows that two events will be able to occur simultaneously in a state s if

1. The preconditions of each event are satisfied in s, and
2. The events are interference-free in s.

It should be noted that, if we wish to show that two events can proceed concurrently (which not only includes simultaneity but also allows either event to precede the other), we also need to prove that the preconditions of each event are independent of the other event. This can be done in the same way as for the correctness conditions. One might, in such circumstances, reserve the term "interference-free" for events that affect neither each other's correctness conditions nor preconditions [8]. Actions $move(A,1)$ and $move(B,1)$ are also interference-free in this stronger sense, as neither action affects the preconditions of the other.

In case two events do affect the same fluents (and thus do not satisfy the condition of interference freedom given above), it might yet be that the fluents are affected in the same way. If this is so, we say that the events are *compatible*. Compatible events can occur simultaneously, and in this sense are also interference-free.

5 Causality

One problem that we have not properly addressed is the apparent complexity of the axioms of independence and correctness. For example, while it might seem reasonable to state that the location of block B is independent of the movement of block A, as everyone knows, this is simply untrue in most interesting worlds. Whether or not the location of B is independent of the movement of A will depend on a whole host of conditions, such as whether B is in front of A, on top of A, on top of A but tied to a door, and so on. Indeed, it is often this apparent endless complexity rather than the combinatorial factors that many people have in mind when they refer to the frame problem.

One way to solve this problem is by introducing a notion of *causality*. We allow two kinds of causation, one in which an event causes the simultaneous occurrence of another event, and the other in which an event causes the occurrence of a consecutive event. We denote these two causal relations by $causes_s(\phi, e_1, e_2)$ and $causes_n(\phi, e_1, e_2)$, respectively, where ϕ is the condition under which event e_1 causes event e_2. These two kinds of causality are sufficient to describe the behavior of any procedure, process, or device that is based on discrete (rather than continuous) events.

The axioms expressing the effects of causation are

$\forall w, s, \phi, e_1, e_2 \,.\, causes_s(\phi, e_1, e_2) \wedge holds(\phi, s) \wedge occurs(e_1, s) \supset occurs(e_2, s)$

$\forall w, s, \phi, e_1, e_2 \,.\, causes_n(\phi, e_1, e_2) \wedge holds(\phi, s) \wedge occurs(e_1, s) \supset occurs(e_2, succ(s, w))$

For example, we might have a causal law to express the fact that, whenever a block x is moved, any block on top of x and not somehow restrained (e.g., by a string tied to a door) will also move. We could write this as

$$\forall x, y, l \,.\, causes_s((on(y, x) \wedge \neg restrained(y)), move(x, l), (move(y, l))$$

If this axiom holds, the movement of x will *cause* the simultaneous movement of y whenever y is on top of x and is not restrained.

We use the notion of causality in a purely technical sense and, while it has many similarities to commonsense usage, we are not proposing it as a fully-fledged theory of causality. Essentially, we view causation as a relation between atomic events that is conditional on the state of the world. We also relate causation to the temporal ordering of events, and assume that an event cannot cause another event that precedes it. However, as stated above, we do allow an event to cause another that occurs simultaneously. This differs from most other formal models of causality [14,20,28], although Allen [1] also allows simultaneous causation.

We can also use causality to maintain invariants over world states and to simplify the specification of actions and events. Consider, for example, a seesaw, with ends A and B and fulcrum F (Figure 1). Assume that A, F, and B are initially at location 0, and consider an event $move_F$ that moves F to location 1. Because of the squareness of the fulcrum and the constraint that A, F, and B must always remain collinear, $move_F$ also results in the movement of A and B to location 1.

We could model $move_F$ by a somewhat complex event that, in and of itself, would affect not only the location of F, but also the locations of A and B. Using this model, the locations of F, A, and B *would not* be independent of $move_F$. However, an alternative view would be to consider that the only property affected by $move_F$ is the location of the fulcrum F, and to let $move_F$ *cause* the simultaneous movement of A and B.

For example, we might have the following causal laws:

$causes_s(true, move_F, move(A, 1))$

$causes_s(true, move_F, move(B, 1))$

The intended meaning of these causal laws is that, if we perform the event $move_F$, both $move(A, 1)$ and $move(B, 1)$ are caused to occur simultaneously with $move_F$.

With this axiomatization, the locations of A and B *will* be independent of $move_F$. Of course, because $move_F$ always causes the movement of A and B, their locations will always be *indirectly* affected by the movement of F; however, from a technical standpoint, we consider the locations of A and B independent of $move_F$ itself (though not of the composite event that consists of the simultaneous movements of F, A, and B). The axioms of independence (and, similarly, correctness) can thus be considerably simplified, but at the cost of introducing more complex causal laws. The advantage of doing things this way is that the complexity is thereby shifted to reasoning about relationships among events, but away from reasoning about the relationships between events and fluents.

There are a number of things to be observed about this approach. First, provided that we add a domain constraint requiring that A, F, and B always remain collinear, we could simplify the above causal laws so that, for example, we require only that $move(A, 1)$ be caused by $move_F$. Using the collinearity constraint, together with the law of persistence, we can then infer that there must exist yet another event that occurs in state s and that brings about the simultaneous movement of B.

In many cases, therefore, we do not need to include causal laws to maintain invariant world conditions; we can instead make use of the constraints on world state to infer the existence of the appropriate events. However, if we do adopt this approach, we shall have to find some way of inferring (either monotonically or nonmonotonically) which of all the potentially appropriate events is the intended one (for example, did the event move B alone, or did it have other effects on the world as well?).

Second, causal laws can be quite complex, and may depend on whether *or not* other events take place as well as on conditions that hold in the world. As a consequence, the application of causal laws need not yield a unique set of caused events. For example, one causal law could require that an event e_1 occur if e_2 does not, whereas another could require that e_2 occur as long as e_1 does not. Given only this knowledge of the world, the most we could infer would be that just one of the events has occurred – but *which* one would be unknown.

Third, for planning possible future courses of action, the extent of causality must somehow be limited. Unless we have either first-order axioms or some nonmonotonic rule to limit causation, any given event could conceivably cause the occurrence of any other event. Clearly, with the possibility of so many events occurring, any useful planning about the future becomes impossible. This is not a difficult problem, but some care must be taken in addressing it [5].

It is interesting to note that the "deductive operators" used in SIPE [31] are very much like the causal laws described herein. Furthermore, SIPE limits the extent of causation by use of an implicit closed-world assumption so that, if causation between any two events cannot be proved, it is assumed that no such relation exists. This yields precisely those events that are *causally necessary*.

Finally, some predicates are better considered as *defined*, which avoids overpopulating the world with causal laws. For example, the distance between two objects may be considered a defined predicate. Instead of introducing various causal laws stating how this relation is altered by various move events, we can simply work with the basic entities of the problem domain and infer the value of the predicate from its definiens when needed.

6 Events as Behaviors

So far, we have identified actions and events with the sets of all their possible behaviors. However, at this point we encounter a serious deficiency in this approach, as well as all others that model actions and events in this way [1,3,20,24,29]. Let us return to the example of the seesaw described in the previous section. Again assume that A, F, and B are initially at location 0, and consider two actions, $move_F$ and $move'_F$, both of which move F to location 1 (see Figure 1). We require that both events also allow all possible movements of A and B, depending on what other events are occurring at the same time (such as someone lifting B). Of course, the objects must always remain collinear. However, $move_F$ and $move'_F$ differ in that, *when performed in isolation*, $move_F$ results in the simultaneous movement of A and B to location 1, whereas $move'_F$ leaves the location of A unchanged while moving B to location 2.

Because both $move_F$ and $move'_F$ exhibit exactly the same class of possible behaviors, the transition relations associated with each of these events will be identical. From a purely behavioral point of view, this is how things should be. To an external observer, it would appear that $move_F$, say, sometimes changed the location of A and not B (when some simultaneous event occurred that raised A to location 2), sometimes changed the location of B and not A (when some simultaneous event raised B), and sometimes changed the locations of both A and B. (Of course, $move_F$ would always change the location of F). As there is no *observation* that could allow the observer to detect whether or not another event was occurring simultaneously, there is no way $move_F$ could be distinguished from $move'_F$ or any other event that had the same transition relation.

On the other hand, it is very convenient to be able to make such distinctions; humans seem to have no trouble reasoning about events of this kind, and it would be unwise to exclude them from our theory. For example, $move_F$ and $move'_F$ may correspond to two different ways of moving F. On the surface, these events would appear to allow the same class of behaviors but, because of unobserved differences in the ways that they are performed, could exhibit different behaviors in specific situations. In other cases, while an event like $move_F$ might be appropriate to seesaws, an event isomorphic to $move'_F$ might be necessary for describing the movement of objects in other contexts. For example, consider the case in which, instead of being components of a seesaw, A is a source of light and B is F's shadow.

Thus, if we wish to be able to distinguish events such as $move_F$ and $move'_F$, we have to allow events with identical transition relations to have differing effects on the world, depending on what other events are, or are not, occurring at the same time. Unfortunately, our current model of events simply leaves us no way to represent this. We cannot restrict the transition relation of $move_F$, say, so that it will always yield the state in which A, F, and B are all at location 1, because that would prevent A or B from being moved simultaneously to other locations. Similarly, we cannot restrict the transition relation of $move'_F$ to allow only the movement of B. Nor can we use any general default rule or minimality criteria to determine the intended effects of an event when performed in any specific context (because that would yield identical models for both $move_F$ and $move'_F$). Indeed, in the situation in which $move_F$ is performed in isolation, note that we do *not* minimize the changes in world relations or maximize their persistence: *both* A and B change location along with F.

We therefore consider events to be objects of the domain that have an associated transition relation, but do not require that events with the same transition relation be deemed equivalent. Events having the same transition relation may differ in other properties; in particular, they may play different causal roles in a theory of the world.[7]

For example, in the case of the seesaw, we might have the following axiom for $move_F$:

$causes_s(p, move_F, move(A, 1))$

where $p \equiv \lambda(s)(\forall e \,.\, occurs(e, s) \supset \; \mathit{int\text{-}free}(e, move(A, 1), s))$.

The intended meaning of this causal law is that the movement of F will cause the movement of A to location 1, provided no event occurs that interferes with the movement of A. If desired, one could use a similar causal law to describe the movement of B.

On the other hand, $move'_F$ would cause the movement of B to location 2 when performed in isolation:

$causes_s(q, move'_F, move(B, 2))$

where $q \equiv \lambda(s)(\forall e \,.\, occurs(e, s) \supset \; \mathit{int\text{-}free}(e, move(B, 2), s))$.

It is important to note that this view of events (and any view that, in one way or another, associates an event with the set of all its possible behaviors [1,3,20,24,29]) requires that the event transition relation include *all* possible transitions under all possible situations, *including* the simultaneous occurrence of other events. For example, if we wish to allow for two pushing events opposed to one another to exert sufficient frictional force on a block to enable it to be lifted, this composite event must be one of the permissible transitions in each of the individual push events. Thus, two events cannot be combined to yield, synergistically, an event that is not part of the actions themselves. It follows that, in the specification of an event, we can only state what is true of *all* possible occurrences of the event. In the case of the push event, it would be a mistake to write an axiom stating that the effect of a push event on a block is to move the block. While this may be true if there is but a single event affecting the block in question, it is clearly false when two or more such events are acting upon the block simultaneously.

What *is* true of all push events (and hence can be stated as an axiom) is that they exert a force on the object being pushed. Other axioms could then be used to specify under what conditions opposing forces generate sufficient frictional force to lift objects, while a third group of axioms could describe how the resultant forces on an object cause it to move. Depending on the desired level of description, this axiomatization could be either simplified or elaborated.

7 Processes

It is often convenient to be able to reason about groups of causally interrelated events as single entitities. For example, we might want to amalgamate the actions and events that constitute the internal workings of a robot, or those that pertain to each component in

[7] In a previous paper [7] I identified events with transition relations and let actions alone assume different causal roles independently of their associated transition relation. Considering the vast literature that already exists on events and actions, and which, if anything, makes a somewhat different distinction, I now believe this to have been a bad idea.

a complex physical system. Such groupings of events, together with the causal laws that relate them to one another, will be called *processes*.

We assume that we have a set of processes and can classify various events and fluents as being either internal or external with respect to these processes. Let $internal_f(\phi, P, s)$ and $external_f(\phi, P, s)$ denote these relationships as they hold between a fluent ϕ and process P in state s, and let $internal_e(e, P, s)$ and $external_e(e, P, s)$ denote these relationships between an event e and process P in state s.

We place a number of constraints on the internal and external fluents and events of a process. First, we require that, for both fluents and events, those classified as internal be mutually exclusive of those classified as external. Second, we require that, in all situations, each internal event have a correctness condition that is dependent only on internal fluents. This property can be expressed as follows:

$$\forall e, s, P \,.\, internal_e(e, P, s) \supset \exists \phi \,.\, internal_f(\phi, P, s) \wedge cc(\phi, e, s)$$

We impose a similar constraint on the preconditions of internal events. Next, we require that internal fluents be independent of all events except internal ones:

$$\forall e, s, \phi, P \,.\, internal_f(\phi, P, s) \wedge \neg internal_e(e, P, s) \supset indep(\phi, e, s)$$

External events and fluents are required to obey similar constraints. It is then not difficult to prove that, if the above axioms are satisfied, the internal events and external events of a given process are interference-free.

Finally, we require that there be no *direct* causal relationship between internal and external events. Thus, the only way the internal events of a given process can influence the external events of the process (or vice versa) is through indirect causation by an event that belongs to neither category. Within concurrency theory, these intermediary events (more accurately, event types) are often called *ports*. Processes thus impose causal boundaries and independence properties on a problem domain, and can thereby substantially reduce cominatorial complexity. Lansky's notion of a *group* [14] is quite similar to our notion of process.

The ease with which processes can be identified will depend strongly on the problem domain. In standard programming systems (at least those that are well structured), processes can be used to represent scope rules and are fairly straightforward to specify. On NASA's proposed space station, most of the properties of one subsystem (such as the attitude control system) will be independent of the majority of actions performed by other subsystems (such as the environmental and life support system), and thus these subsystems naturally correspond to processes as defined here. Lansky [14] gives other examples in which processes are readily specified.

In other situations, the specification of processes might be more complicated. For example, we might know that interference at a distance can occur only as a result of electromagnetic or gravitational phenomena, and so utilize this knowledge to impose causal boundaries whenever electromagnetic emissions are shielded and gravitational forces are negligible. Moreover, in many real-world situations, dependencies will vary as the spheres of influence and the potential for interaction change over time. For example, consider how many actions in real life are taken solely for the purpose of limiting or enhancing interference with other systems (such as closing a door for privacy, camouflaging military equipment, or making a phone call).

If we are to exploit the notion of process effectively, it is important to define various composition operators and to show how properties of the behaviors of the composite processes can be determined from the behaviors of the individual processes. For example, we should be able to write down descriptions of the behaviors of individual agents, and from these descriptions deduce properties of groups of agents acting together (concurrently). We should *not* have to consider the internal behaviors of each of these agents to determine how the group as a whole behaves. In contrast, all the so-called hierarchical planning systems (e.g., NOAH [27] and SIPE [31]) analyze interaction down to the atomic level (as was noticed early by Rosenschein [26]).

The existing literature on concurrency theory [13,21] provides a number of useful composition operators. Some examples are given below. They can all be defined in terms of the causal relations introduced earlier, although we need to introduce a special "no-op" or "wait" event to prevent forcing the processes to operate in lockstep with one another.

Prefixing (:)

The process $e : P$ is one that can begin by performing the event (strictly speaking, event *type*) e, after which it behaves exactly like P.

Sequencing (;)

The process $P \ ; \ Q$ behaves first like P and, if that concludes successfully, behaves next like Q.

Ambiguity (+)

The process $P + Q$ can behave like either P or Q. For example, if

$R = (b : P) + (c : Q) \quad ,$

then R can either perform b and evolve into P or perform c and evolve into Q.

Parallelism (&)

The process $P \ \& \ Q$ is one in which both P and Q run concurrently. Events that are designated as synchronous must occur simultaneously, whereas other events can choose to occur simultaneously with one another or be arbitrarily interleaved.

We must, of course, provide various axioms about these operators so that they will be useful. For example, it is not difficult to show that any [temporal] property that holds of the behaviors of a process P (or Q) will also hold for the composite process $P \ \& \ Q$.

Such axioms may not appear to be very useful, as the properties that hold of each process will, in general, depend on what other events could occur in the environment. However, to the extent that the properties of a process's behaviors are specified in terms of its internal events and fluents, they will be independent of the context in which the process is embedded. For example, consider the two very simple processes P and Q given below:

$P = a : b : P$, and

$Q = c : d : Q$.

Strictly speaking, we have to provide a fixed-point operator to define these processes [13], but the intended meaning should be clear. Now, if the event types a and b are mutually exclusive and are internal with respect to process P, all behaviors of P will be such that the number of events of type a, denoted $\#a$, and the number of events of type b, $\#b$, obey the constraint: $\#a - 1 \leq \#b \leq \#a$. This would *not* be the case if a and b were not internal events of process P, as any other process could then choose to perform an arbitrary number of events of type a or b concurrently with process P.

Now let's assume that c and d are internal events of process Q, and thus obey a law similar to that given for P. Furthermore, let us assume that events of type b and d are always constrained to be simultaneous with some interface event e, and vice versa. Using the fact that the behaviors of the composite process P & Q will satisfy the same constraints as the component processes, it is not difficult to prove that $\#a - 1 \leq \#c \leq \#a + 1$.

Despite the triviality of the example, the important point is that, in proving the above result, we have not had to examine the internal workings of either process P or Q. For example, had the internal structure of these processes been entirely different, this result would have remained valid (provided, of course, that the processes had still imposed the same constraints on the number of occurrences of events a, b, c, and d). Furthermore, in determining the properties of the individual behaviors of P and Q we have not had to consider the external environment in which they are embedded; the relationship between the number or occurrences of events a and b in process P is independent of external happenings, and similarly for events c and d in process Q.

There are a number of complexities in the specification of processes that require some care. For example, a process may *fail* at any time (because some precondition has not been or cannot be met, or some correctness condition has been violated, etc.). Thus, the behaviors generated by a process will include both failed and successful behaviors, where each failed behavior is a prefix of some successful behavior. Indeed, the notion of a process generating both successful and failed behaviors is essential to planning in real-world domains – we frequently select a plan of action according to how it can *fail* rather than how it succeeds. Amy Lansky and I have discussed this question in more detail elsewhere [9].

Nondeterministic processes present another problem that has to be addressed with caution. Such processes can differ from one another despite the fact that they generate identical behaviors (both successful and failed). The reason for this is that nondetermistic processes can behave differently in different environments even though their sets of potential behaviors are identical. This issue is explored at length in the literature on concurrency and various means of handling the problem have been developed [13,21].

In summary, the notion of process allows us to structure problem domains and thus avoid considering how every event affects every other event or fluent. It can therefore lead to a substantial reduction in the combinatorics of the problem. Furthermore, we can describe complex domains in a compositional way – that is, we can specify properties of individual processes independently of other processes, and then determine the behavior of the combined processes from knowledge of the behaviors of the component processes.

8 The Frame Problem

The frame problem, as Hayes [12] describes it, is concerned with providing, in a reasonably natural and tractable way, appropriate "laws of motion." Our approach to this problem has been to provide a proper model-theoretic account of actions and events, and to formulate first-order axioms and rules that allow the effects of actions and events to be determined monotonically. Furthermore, by introducing the notions of independence and correctness, we obviate the necessity of using nonmonotonic operators or consistency arguments to obtain useful results regarding persistence and interference.

Of course, we are left with the problem of *specifying* independence and correctness. There are essentially two problems here: (1) the apparent combinatorial difficulties in expressing all the required independence and correctness axioms; and (2) the complexity to be expected of many, if not most, of these axioms in real-world applications.

The first of these problems is probably overstated in much of the literature on the frame problem. Thus, just as in the axiomatization of any large or infinite domain, the number of axioms required to specify independence (and correctness) can be substantially reduced by the use of general axioms that allow specific instances of independence to be deduced as needed. This can substantially reduce the combinatorial problem and has the advantage of remaining within the bounds of first-order logic.

Where desired, some simple closed-world assumption or minimality criterion could additionally be used to ease specification of the independence relation for the basic fluents of the domain. This can be quite straightforward, although some care has to be taken with the situation variable [5,10].

The second problem is handled by introducing causal laws that describe how actions and events bring about (cause) others. Indeed, without such causal laws, the specification of independence would become as problematic as is the specification of persistence in the standard formalisms. Of course, the causal laws can themselves be complex (just as is the physics of the real world), but the representation and specification of actions and events are thereby kept simple.

Some researchers take a more general view of the frame problem, seeing it as the problem of reasoning about the effects of actions and events with *incomplete information* about what other events or processes (causally related or otherwise) may also be occurring. Unfortunately, this problem is often confused with that of providing an adequate *model* of events, with the result that there is usually no clear model-theoretic semantics for the representation.

For example, one of the major problems in reasoning about actions and plans is to determine which events can possibly occur at any given moment. Based on the relative infrequency of "relevant" events (or that one would "know about" these if they occurred), it has been common to use various default rules [25], nonmonotonic operators [3,20], or minimal models [15,18,28] to constrain the set of possible event occurrences. However, there are many cases in which this is unnecessary – where we can *prove*, on the basis of general axioms about independence and correctness, that no events (or effects) of interest could possibly occur. We may even have axioms that allow one to avoid considering whole classes of events, such as when one knows that certain events are external with respect to a given process. Thus, in many cases, there may be no need to use default rules or minimality principles – reasoning about plans and actions does not have to be nonmonotonic.

When we *do* need to make assumptions about event occurrences, default rules and circumscription can be very useful. For example, by minimizing the extensions of the *occurs* and *causes* predicates, we can obtain a theory in which the only event occurrences are those that are *causally necessary* [5]. This kind of reasoning seems to correspond closely to much of commonsense planning. However, in making reasonable assumptions about a given domain, we do not have to limit ourselves to such default rules or minimality criteria. In some cases, it may be preferable to use, for example, domain-specific rules defining what assumptions are appropriate or, alternatively, a more complicated information-theoretic approach based on quantitative probabilities of event occurrences.

To take a familiar example [1], it seems reasonable, at first, to assume that my car is still where I left it this morning, unless I have information that is inconsistent with that assumption. However, this premise gets less and less reasonable as hours turn into days, weeks, months, years, and centuries – even if it is quite *consistent* to make such a premise. This puts the problem where it should be – namely, in the area of making reasonable assumptions, not in the area of *defining* the effects of actions [4,12,25], the persistency of facts [3,20], or causal laws [28].

Of course, most of the axioms we might state about the real world are subject to *qualification* [19]. Our aim has not been to solve this problem, although the notion of causality helps to some extent. Furthermore, because our approach has a well-defined semantics and can be formalized in first-order logic, we provide a sound base on top of which can be built various meta-theories regarding the handling of qualifications and other kinds of assumptions. I address some of these issues elsewhere [5].

9 Conclusions

We have constructed a model of actions and events suited to reasoning about domains that involve multiple agents or dynamic environments. The proposed model provides for simultaneous actions, and a generalized situation calculus is used to describe the effects of actions in multiagent settings. Notions of *independence* and *correctness* were introduced and it was shown how they can be used to determine the persistence of facts over time and whether or not actions can be performed concurrently. Both these notions I consider critical to reasoning *effectively* about multiagent domains. Furthermore, unlike most previous formalisms in both single and multiagent domains, the proposed law of persistence is *monotonic* and therefore has a well-defined model-theoretic semantics.

We have also demonstrated how the concept of *causality* can be used to simplify the description of actions and to model arbitrarily complex machines and physical devices. It was also shown how sets of causally interrelated actions can be grouped together in *processes* and how this structuring of a problem domain can substantially reduce combinatorial complexity. The notion of structuring the problem domain by using general axioms about independence and causal influence appears to be essential for solving complex multiagent problems. The only existing work I know of that incorporates such an idea is the planning system being developed by Lansky [14].

Although we did not consider implementation issues directly, the concepts and laws introduced by us were aimed at providing a sound basis for practical planning and reasoning systems. For example, one of the most efficient action representations so far employed in AI planning systems – the STRIPS representation [4,15] – is essentially the special case in which (1) the effects of an action can be represented by a conjunction of either positive or negative literals, called the *add list*; (2) the action is independent of all properties except those given in (or deducible from) the *delete list* of the action; (3) a single precondition determines performabilty of the action; and (4) no actions ever occur simultaneously with any other. The approach used by Pednault [23] can also be considered the special case in which there are no simultaneous actions.

Furthermore, the work here indicates how the STRIPS representation could be extended to the multiagent domain. For example, one possibility would be to stay with the single-agent representation, adding to it the requirement that the correctness conditions of an action be represented by the same conjunction of literals given in the add list of the action. Causal laws could be introduced in the manner of the deductive operators of SIPE [31], thereby increasing the expressive power of the approach without introducing the problems usually associated with extending the STRIPS assumption [25]. In this way, some of the traditional planning systems may be able to be modified to handle multiagent domains. Alternatively, the approach of Manna and Waldinger [17] could be applied to these domains by employing the generalized situation calculus we introduced here. Finally, our approach has shown how we can deduce constraints on the occurrence of actions and events from relatively simple axioms about their effects and influence. These constraints can then be used by event-based planners [6,14,30] to form synchronized plans involving the cooperation of multiple agents in dynamically changing environments.

9.1 Acknowledgments

I wish especially to thank Amy Lansky and Ed Pednault, both of whom helped greatly in clarifying many of the ideas presented in this paper. I am also indebted to Vladimir Lifschitz for some very enlightening discussions, and to Richard Waldinger who criticized an earlier draft of this paper.

References

[1] J. F. Allen. *A General Model of Action and Time.* Technical Report 97, University of Rochester, Rochester, New York, 1981.

[2] M. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, Cambridge, Massachusetts, forthcoming.

[3] T. Dean. Planning and temporal reasoning under uncertainty. In *Proceedings of the IEEE Workshop of Knowledge-Based Systems*, Denver, Colorado, 1984.

[4] R. E. Fikes and N. J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[5] M. P. Georgeff. Actions, processes, and causality. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, Timberline Lodge, Timberline, Oregon, 1987.

[6] M. P. Georgeff. Communication and interaction in multiagent planning. In *Proceedings of the Third National Conference on Artificial Intelligence*, pages 125–129, Washington, D.C., 1983.

[7] M. P. Georgeff. The representation of events in multiagent domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 1986.

[8] M. P. Georgeff. A theory of action for multiagent planning. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, Austin, Texas, 1984.

[9] M. P. Georgeff and A. L. Lansky. *A System for Reasoning in Dynamic Domains: Fault Diagnosis on the Space Shuttle.* Technical Note 375, Artificial Intelligence Center, SRI International, Menlo Park, California, 1986.

[10] S. Hanks and D. McDermott. Default reasoning, nonmonotonic logics, and the frame problem. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, 1986.

[11] D. Harel. *First Order Dynamic Logic. Lecture Notes in Computer Science, 68*, Springer-Verlag, New York, New York, 1979.

[12] P. J. Hayes. The frame problem and related problems in artificial intelligence. In Elithorn A. and D. Jones, editors, *Artificial and Human Thinking*, pages 45–59, Jossey-Bass, 1973.

[13] C. A. R. Hoare. *Communicating Sequential Processes. Series in Computer Science*, Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[14] A. L. Lansky. *Localized Representation and Planning Methods for Parallel Domains.* Technical Note 401, Artificial Intelligence Center, SRI International, Menlo Park, California, 1986.

[15] V. Lifschitz. *Circumscription in the Blocks World.* Computer Science Working Memo, Stanford University, Stanford, California, 1985.

[16] V. Lifschitz. On the semantics of STRIPS. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, Timberline Lodge, Timberline, Oregon, 1987.

[17] Z. Manna and R. J. Waldinger. A theory of plans. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, Timberline Lodge, Timberline, Oregon, 1987.

[18] J. McCarthy. Circumscription – a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[19] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[20] D. McDermott. *A Temporal Logic for Reasoning about Plans and Processes.* Computer Science Research Report 196, Yale University, New Haven, Connecticut, 1981.

[21] R. Milner. *A Calculus of Communicating Systems. Lecture Notes in Computer Science 92*, Springer Verlag, New York, New York, 1980.

[22] E. P. D. Pednault. Solving multiagent dynamic world problems in the classical planning framework. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, Timberline Lodge, Timberline, Oregon, 1987.

[23] E. P. D. Pednault. *Toward a Mathematical Theory of Plan Synthesis.* PhD thesis, Department of Electrical Engineering, Stanford University, Stanford, California 94305, forthcoming.

[24] R. Pelavin. *A Formal Logic for Planning with a Partial Description of the Future.* PhD thesis, Department of Computer Science, University of Rochester, Rochester, New York, forthcoming.

[25] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[26] S. J. Rosenschein. Plan synthesis: a logical perspective. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 331–337, Vancouver, British Columbia, 1981.

[27] E. D. Sacerdoti. *A Structure for Plans and Behaviour.* Elsevier, North Holland, New York, 1977.

[28] Y. Shoham. Chronological ignorance: time, nonmonotonicity, necessity and causal theories. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 389–393, Philadelphia, Pennsylvania, 1986.

[29] Y. Shoham. Ten requirements for a theory of change. *New Generation Computing*, 3:467–477, 1985.

[30] C. J. Stuart. *Synchronization of Multiagent Plans Using A Temporal Logic Theorem Prover.* Technical Note 350, Artificial Intelligence Center, SRI International, Menlo Park, California, 1985.

[31] D. E. Wilkins. Domain independent planning: representation and plan generation. *Artificial Intelligence*, 22:269–301, 1984.